https://doi.org/10.31649/1997-9266-2026-184-1-159-166

УДК 004.75

О. В. Войцеховська[1]
Р. Ю. Чехместрук[1]
Д. В. Марценюк[1]
О. В. Войцеховський[2]

СИСТЕМА АВТОМАТИЧНОГО АВАРІЙНОГО ВІДНОВЛЕННЯ ІНФОРМАЦІЙНИХ ВЕБРЕСУРСІВ У ХМАРНОМУ СЕРЕДОВИЩІ

[1]Вінницький національний технічний університет;
[2]Київський національний університет імені Тараса Шевченка

*Проаналізовано сучасні підходи до аварійного відновлення (*Disaster Recovery) *вебдодатків у хмарних середовищах та гібридних інфраструктурах. Розглянуто основні моделі Disaster Recovery, такі як Backup-and-Restore, Warm Standby, Multi-AZ та Multi-Region, а також їхні переваги й обмеження з погляду вартості реалізації, швидкості відновлення та вимог до обчислювальних ресурсів. Особливу увагу приділено сценаріям, у яких локальні сервери поєднуються з хмарними сервісами Amazon Web Services, що дають можливість досягнути балансу між економічністю рішення та високою відмовостійкістю.*

Досліджено механізми автоматизації процесів відновлення, зокрема роль Amazon Route 53 Health Check, CloudWatch Alarm, SNS та Lambda як тригерів і основних компонентів DR-алгоритму.

Розроблено систему автоматичного аварійного відновлення вебресурсів у гібридному середовищі AWS, яка базується на автоматичному перемиканні на резервне середовище та вибірковому відновленні лише критично важливих компонентів системи. Побудовано математичні моделі оцінювання показників RTO та RPO, які враховують часові затримки на виявлення збоїв, відновлення бази даних, запуск застосунку та оновлення DNS-маршрутів. Розроблено підходи до відновлення даних на основі періодичного резервного копіювання у S3 та безперервної реплікації за допомогою AWS Database Migration Service. Доведено, що традиційні підходи, які потребують повного дублювання інфраструктури, є ресурсомісткими та недостатньо гнучкими для умов з динамічно змінним рівнем доступності локальних ресурсів.

Виконано розгортання тестового вебресурсу у хмарному середовищі AWS. Спроектовано високодоступну інфраструктуру з використанням сервісів Amazon, усі компоненти якої створено з використанням підходу Infrastructure as Code засобами Terraform, що забезпечує повторюваність, керованість і можливість швидкого розгортання рішень у різних середовищах.

За результатами проведеного моделювання та тестування системи встановлено суттєве скорочення часу простою вебресурсу у разі застосування запропонованої системи аварійного відновлення порівняно з традиційними DR-стратегіями.

Ключові слова: аварійне відновлення, AWS, вебресурс, гібридна інфраструктура, реплікація, Route53, CloudWatch, Lambda, RTO, RPO.

Вступ

В сучасному суспільстві спостерігається стрімке зростання різного роду цифрових сервісів, які охоплюють майже всі сфери діяльності людини і є основою функціонування бізнес-процесів, інфраструктурних компонентів, систем електронної комерції, державних порталів, освітніх платформ тощо. Будь-який збій в роботі цих сервісів, незалежно від його природи, може спричинити суттєві фінансові та репутаційні втрати. Це зумовлює необхідність у створенні високонадійних інформаційних систем, здатних автоматично реагувати на будь-які технічні збої та забезпечувати безперервність роботи критично важливих вебресурсів.

Часткова або повна втрата працездатності вебсервісів може бути спричинена апаратними від-

мовами, збоєм мережного обладнання, вразливостями у програмному забезпеченні, людським фактором або кібератаками, які в останні роки мають тенденцію до стрімкого зростання. Крім апаратних ризиків, значний вплив мають і природні чинники, такі як стихійні лиха, лісові пожежі або глобальні аварії на рівні інженерних систем. За даними міжнародних досліджень, кожна година простою високонавантаженої інформаційної системи для більшості середніх і великих підприємств може коштувати більше 300 тисяч доларів [1].

Згідно з опитуванням [2], проведеного компанією Dimensional Research, 76 % IT-фахівців повідомили про серйозну втрату критично важливих даних у своїй організації, а 45 % з них зазнали безповоротної втрати даних. При цьому 95 % опитуваних вказали, що їхня компанія має план аварійного відновлення, проте лише у 24 % такий план є якісно задокументованим, перевіреним та вчасно оновлюваним.

У зв'язку з цим ефективна стратегія аварійного відновлення Disaster Recovery (DR) стає не лише технічною необхідністю, а й важливим елементом управління ризиками та убезпечення бізнес-процесів.

Особливої актуальності питання аварійного відновлення вебресурсів набуває в Україні, де рівень технологічних і зовнішніх загроз є значно вищим порівняно з багатьма країнами. Енергетична нестабільність, перебої з електропостачанням, атаки на критичну інфраструктуру, а також військові дії створюють додаткові ризики повного виходу з ладу локальних серверів та інформаційних систем. У таких умовах використання виключно локальних обчислювальних ресурсів стає ненадійним і потребує переходу до гібридних архітектур, у яких локальні середовища доповнюються хмарною інфраструктурою з високим рівнем відмовостійкості [3].

Гібридні моделі дозволяють поєднувати переваги локальних рішень (контроль над даними, мінімальні затримки, можливість роботи у закритих мережах) з можливостями хмарних сервісів — масштабованістю, розподіленістю, високою доступністю та автоматизованими механізмами реагування на збої. Саме хмарні платформи, надають потужний інструментарій для побудови систем аварійного відновлення, зокрема, механізми моніторингу, реплікації, маршрутизації трафіку та автоматичного запуску резервних компонентів.

Метою роботи є створення такої системи автоматичного аварійного відновлення вебресурсів, яка забезпечить мінімальний час простою, буде здатною адаптуватися до різних сценаріїв роботи, враховуючи реальні технологічні та економічні обмеження конкретної організації.

Огляд сучасних DR-підходів аварійного відновлення

Сучасні системи аварійного відновлення (DR-системи) у хмарних середовищах базуються на різних рівнях відмовостійкості. Кожен рівень характеризується компромісом між швидкістю відновлення та фінансовими витратами.

Найпростішою моделлю Disaster Recovery є модель Backup-and-Restore, суть якої полягає у тому, що дані захищаються шляхом їх копіювання та збереження у безпечному місці з можливістю подальшого відновлення у випадку аварії або збою. Проте вона передбачає значний час простою, оскільки потребує повного відновлення даних з резервної копії і, отже, придатна лише для систем, де довготривалі перерви не є критичними.

Модель Pilot Light зберігає мінімально необхідний набір ресурсів у хмарі, що значно зменшує максимальний час, протягом якого система може не працювати після збою. Вона підходить для сервісів із середнім рівнем критичності, де немає необхідності в миттєвому перемиканні.

Модель Warm Standby підтримує частково активну копію системи. Це забезпечує досить швидке відновлення, але потребує постійних витрат на підтримку інфраструктури.

Метод з використанням багатьох зон доступності Multi-AZ гарантує синхронну реплікацію даних між зонами доступності. Такий підхід забезпечує нульовий обсяг даних, які можна втратити під час збою, але є дорогим для невеликих організацій.

Метод розташування даних в багатьох географічних регіонах хмарної інфраструктури Multi-Region забезпечує географічне резервування, але його впровадження є найдорожчим. Така архітектура використовується лише у глобальних інфраструктурних рішеннях [4].

Отже, проведений аналіз показав, що найефективнішим рішенням є використання гібридних архітектур, які дозволяють поєднати локальні ресурси, хмарні резерви та автоматизацію перемикання.

Архітектура системи автоматичного аварійного відновлення вебресурсів

В роботі запропоновано систему автоматичного аварійного відновлення вебресурсів у хмарному середовищі з використанням гібридної архітектури розгортання на платформі Amazon Web Services, в якій відбувається відновлення критичних ресурсів замість відновлення всієї системи, а процес аварійного відновлення здійснюється автоматично без ручного втручання.

Архітектура розробленої системи, схему якої показано на рисунку, містить чотири основних компоненти: AWS Route 53, Amazon CloudWatch, Amazon SNS, AWS Lambda.

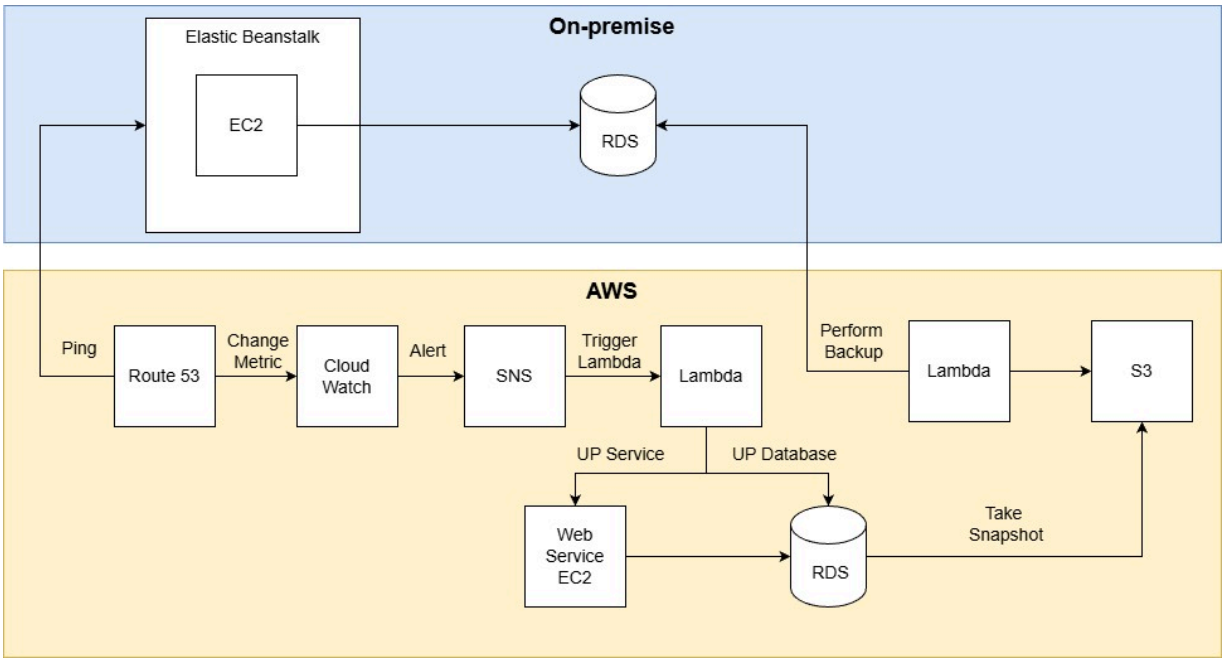

Схема архітектури методу аварійного відновлення

Компонент AWS Route 53 за допомогою механізму health check виконує безперервний контроль доступності локального ресурсу, надсилаючи DNS-запити до нього, на які, ресурс надсилає відповіді [5]. Якщо Route 53 три рази (заданий поріг відмови) поспіль не отримав відповідь на запит, то стан ресурсу переходить у значення Unhealthy, який свідчить про недоступність основного середовища.

Тоді, оскільки налаштовано сповіщення для перевірки справності ресурсу, сервіс CloudWatch отримує агреговані метрики та формує сповіщення про необхідність активації алгоритму відновлення, яке передається до сервісу доставки повідомлень SNS.

У свою чергу SNS запускає Lambda-функції, які є обчислювальним ядром системи та містять основні механізми логіки аварійного відновлення. Lambda ініціює алгоритм аварійного відновлення, адаптований під вибрану стратегію відновлення, та виконує більше десятка операцій: створення нового екземпляра RDS, імпорт CSV-файлів, перевірку стану таблиць, запуск Auto Scaling Group, оновлення DNS та валідацію працездатності відновленого вебресурсу. Такий підхід гарантує повну автоматизацію процесу. Отже, у випадку збою локального сервера, система автоматично перемикає навантаження на резервну інфраструктуру в AWS.

Для аналізу ефективності будь-якої системи аварійного відновлення необхідне формалізоване визначення показників, які відображають реальну здатність системи протидіяти збоям та забезпечувати безперервність роботи.

В системах Disaster Recovery для визначення їх ефективності використовують два основних показники — цільовий час відновлення RTO (Recovery Time Objective), тобто максимальний час, протягом якого система може не працювати після збою та цільова точка відновлення RPO (Recovery Point Objective), тобто максимально допустимий обсяг даних, які можна втратити. Ці показники дозволяють кількісно оцінити очікуваний час відновлення й максимально допустимий обсяг втрати даних [6].

Ці показники є базовими для вибору архітектурних рішень, визначенні рівня готовності резервної інфраструктури, встановленні частоти резервного копіювання та оцінюванні економічної доцільності різних DR-стратегій. Разом з тим їхні значення залежать від багатьох факторів, зокрема

швидкості виявлення відмови, особливостей алгоритму перемикання трафіку, типу відновлення бази даних (холодне або гаряче), пропускної здатності мережі, продуктивності хмарних сервісів та тривалості розповсюдження DNS-змін.

Оскільки запропонована система аварійного відновлення включає як елементи повного відновлення системи з резервних копій, так і сценарії з безперервною реплікацією, створено математичну модель, яка відображає часову структуру всього процесу переходу на резервне середовище. Модель дозволяє визначити залежності між окремими етапами відновлення та вплив кожного компонента (виявлення збою, активація інфраструктури, імпорт чи реплікація даних, оновлення мережевих маршрутів) на загальний час відновлення.

Максимально допустимий час *RTO*, протягом якого вебресурс може бути недоступним після аварії, визначає, скільки часу має пройти від моменту збою до повного відновлення працездатності сервісу

$$RTO = t_{restore} - t_{failure}, \tag{1}$$

де $t_{restore}$ — момент реєстрації відмови; $t_{failure}$ — момент повного відновлення працездатності системи.

Максимально допустима втрата даних *RPO*, виміряна у часі, показує, за який проміжок часу дані можуть бути втрачені без критичної шкоди для бізнесу

$$RPO = t_{restore} - t_{last\_backup}, \tag{2}$$

де t_{last_backup} — момент останнього бекапу.

Визначення параметрів *RTO* (1) та *RPO* (2) забезпечує можливість порівняння різних підходів аварійного відновлення, включно з традиційними сценаріями Multi-AZ, Warm Standby, Backup-and-Restore та запропонованою гібридною архітектурою. Це дозволяє не лише оцінити поточну модель, але й оптимізувати її шляхом зміни інтервалів резервування, підвищення частоти Health-Check перевірок або використання більш продуктивних механізмів відновлення.

Для алгоритму роботи, коли відновлення бази даних виконується з CSV-файлів у S3 (холодна модель), значення максимального часу недоступності вебресурсу *RTO* суттєво залежить від обсягу таблиць, швидкості імпорту та затримок між сервісами S3 і RDS.

Ще одним важливим параметром є інтервал між моментом фактичної відмови сервера та моментом, коли механізм контролю стану зафіксував, що сервіс став непрацездатним:

$$FTD = t_{detected} - t_{crash}, \tag{3}$$

де $t_{detected}$ — момент виявлення збою системою; t_{crash} — момент фактичної відмови або виникнення збою в системі/сервері.

Мінімізація часу *FDT* є важливою, оскільки вона безпосередньо прискорює запуск процесу відновлення.

Загальний час, необхідний для відновлення системи після збою *TRT* визначається за формулою [7]

$$TRT = FDT + RTO. \tag{4}$$

На основі отриманого значення *TRT* можна оцінити рівень доступності системи за моделлю Service Level Agreement (*SLA*), що є одним із найважливіших показників якості роботи сервісів і безпосередньо визначає рівень надійності інфраструктури. Вимірюється у відсотках часу, протягом якого система залишається працездатною та доступною для користувачів.

У системах з гібридною архітектурою момент фіксації збоїв та момент відновлення не часто є миттєвими. Зокрема, час виявлення збою *FDT* суттєво визначається частотою виконання health check-запитів у Route 53.

Для перевірки доступності ресурсу у Route 53 вибрано інтервал 30 секунд, оскільки він забезпечує оптимальний баланс між швидкістю виявлення відмови та стабільністю роботи системи. Такий інтервал дозволяє достатньо швидко ініціювати сценарій Disaster Recovery у разі реального збою, водночас зменшуючи ймовірність хибних спрацьовувань через короткочасні мережеві збої.

З урахуванням того, що процес DR містить запуск Lambda-функції, ініціалізацію резервної бази даних та розгортання застосунку в AWS, загальний час відновлення складається не лише з інтервалу перевірки, а і з часу виконання цих операцій. Тому інтервал 30 секунд не створює критичного впливу на RTO, але забезпечує достатньо швидке реагування системи на реальну відмову.

Така конфігурація дозволяє зменшити зайве навантаження на механізм health check та зберегти

стабільність роботи всієї архітектури, що особливо важливо у разі автоматичного запуску аварійного сценарію без участі адміністратора.

Отже, якщо інтервал перевірки становить 30 секунд, а поріг кількості невдалих відповідей дорівнює трьом, система гарантовано виявить збій у діапазоні від 30 до 90 секунд. Це означає, що компонент *FDT* у формулі (3) є статистично варіативним і на практиці розглядається як інтервал, а не фіксована величина.

Отже, параметри *RTO* та *RPO* дозволяють визначити межі застосування запропонованої системи аварійного відновлення та виявити фактори, що найбільше впливають на швидкість і повноту відновлення ресурсу після аварії.

Проведене тестування холодної моделі розробленої системи показало лінійну залежність часу відновлення від обсягу набору даних. Це означає, що зі збільшенням обсягу критичних таблиць на 20...30 % загальний *RTO* зростає пропорційно, якщо не застосовуються додаткові оптимізації, такі як паралельний імпорт або попереднє створення індексів.

Для забезпечення зменшення максимального часу недоступності вебресурсу *RTO* запропоновано модифікацію розробленої системи (гаряча модель) з використанням AWS Database Migration Service (DMS) [8], в якій використовується CDC-реплікація, яка дозволяє миттєво перемикатися на резервний екземпляр без необхідності імпортування даних.

До того ж, значення *RTO* для гарячої моделі здебільшого визначається етапами активації резервного середовища, а не відновленням інформації. Оскільки дані вже перебувають у синхронізованому стані, тривалість відновлення залежить переважно від запуску інфраструктури Auto Scaling Group, оновлення DNS-записів та короткої перевірки доступності застосунку. На практиці це забезпечує зменшення часу простою у 3—5 разів у порівнянні з холодним підходом.

Тестування системи та порівняння її з альтернативними DR-рішеннями

Для перевірки ефективності запропонованої системи автоматичного аварійного відновлення проведено тестування на експериментальному стенді з параметрами, максимально наближеними до реальних умов роботи вебдодатку.

Як тестовий, використано NET Core MVC вебдодаток для тестування та організації навчального процесу, розгортання якого виконано шляхом інтеграції середовища розробки Microsoft Visual Studio з інструментарієм Amazon SDK for .NET, що містить плагіни для публікації застосунків у хмару.

Розмір бази даних у випробуваннях становив близько 220 МБ і включав 14 критично важливих таблиць загальним обсягом понад 180 тисяч записів. Найбільшою за обсягом була група таблиць, пов’язаних з результатами тестувань, де зберігалося понад 150 тисяч рядків, тоді як у таблиці користувачів налічувалося близько 12 тисяч записів, а у структурних таблицях дисциплін та занять — понад 4,5 тисяч.

Для імітації реального навантаження під час тестів використовувалася модель одночасної роботи 50 активних користувачів, яка створювала середню інтенсивність у межах 250...300 HTTP-запитів за секунду. У потік звернень входили операції автентифікації, завантаження навчальних матеріалів, виконання тестових завдань та запити до довідкової інформації. Такий характер навантаження дозволив відтворити типовий сценарій пікової активності в освітніх та корпоративних системах. Аналіз виконувався за такими критеріями: час відновлення *RTO*, втрата даних *RPO*, фінансові витрати на утримання інфраструктури, гнучкість налаштування та придатність до гібридних сценаріїв, де частина системи працює локально.

Параметри відновлення бази даних для різних варіантів аварійного відновлення подано в таблиці.

Результати порівняння показали, що запропонований метод є у 1,5—3 рази економнішим ніж використання AWS Elastic Disaster Recovery та Multi-AZ. Це пояснюється тим, що класичні рішення AWS передбачають постійне дублювання обчислювальних ресурсів у режимі реального часу, що призводить до значного фінансового навантаження. Натомість гібридний підхід дозволяє підтримувати у хмарі лише резервні компоненти, необхідні для швидкого відновлення, без повного дублювання всієї інфраструктури, що зменшує вартість обслуговування у 2—4 рази.

Порівняння з Azure Site Recovery підтвердило переваги запропонованого рішення у сценаріях, де локальні сервери відіграють ключову роль. Azure ASR орієнтований переважно на повну міграцію або реплікацію віртуальних машин у хмару, що є ефективним у середовищах типу «cloud-first», проте менш гнучким у випадках з локально-орієнтованими системами. У таких випадках гібридна

архітектура AWS демонструє кращу інтегрованість, простіший механізм автоматичного перемикання та менший поріг входу в експлуатацію.

Порівняння параметрів аварійного відновлення бази даних

Характеристика	Холодна модель CSV + S3	Гаряча модель DMS-реплікація	Аналог Availability Zones	Аналог Elastic Disaster Recovery	Аналог Azure Site Recovery
RTO (час відновлення), хв	7	1,5	≤ 1	≈15	≈15
RPO (втрата даних) , хв	5	≈ 0	≤ 1	1...5	1...5
FDT (виявлення відмови), с	30	30	20...30	20...30	30...60
TRT (повний час простою) , хв	≈ 7	≈ 1,5	≈ 1	≈ 16	≈ 16
SLA (оцінка доступності), %	99,999	99, 9999	99,9999	99,99	99,99
Вартість інфраструктури	нижча (оплата лише за S3 та використання Lambda)	вища (DR-база завжди працює)	вища (реплікація в межах регіону)	вища (повні бекапи серверу)	вища (повні бекапи серверу)
Складність реалізації	проста: експорт/імпорт CSV, автоматизація через Lambda	середня: налаштування DMS, CDC, автоматичне перемикання	середня	проста	середня: складні мережеві сценарії
Придатність для DR	придатна для економних систем та систем, нечутливих до втрати останніх даних	оптимальна для критичних систем з мінімальним RTO/RPO	для глобальних систем з нульовим простоєм	для компаній, які хочуть отримати просте рішення	для компаній, що працюють в Azure
Ціна	$3...7	$45...75	$200... 320	$120...150	$140...200

Водночас проведено порівняння холодної та гарячої моделей відновлення даних.

У холодній моделі *RPO* визначається інтервалом резервного копіювання (експорт CSV у S3), тому система може втратити до 5 хвилин даних. Такий підхід придатний для інформаційних систем, де короткочасна втрата не є критичною.

Натомість гаряча модель з AWS DMS забезпечила практично нульовий *RPO*, оскільки синхронізація відбувається у режимі Change Data Capture. Це особливо важливо для систем з високими вимогами до збереження транзакційних даних, наприклад фінансових або логістичних платформ. Отримано, що *RTO* у гарячому сценарії становить 1...2 хвилини, тоді як холодний підхід дає близько 6...7 хвилин.

Для порівняння, усереднений *RTO* для Azure ASR може перевищувати 10 хвилин, а для Elastic Disaster Recovery— коливатися в діапазоні 3...5 хвилин залежно від розміру інфраструктури. Отже, за швидкістю відновлення запропоноване гібридне рішення з AWS Lambda та DMS є одним із найбільш конкурентоспроможних.

До того ж, проведене порівняння показало, що час повного простою *TRT* скоротився на 30...50 %, витрати інфраструктури на 40...60 % нижчі, а також, що за використання холодного методу відновлення рівень доступності склав близько 99,9999 %, тоді як гарячий підхід з постійною DMS-реплікацією забезпечує 99,99999 %, що фактично відповідає режиму високої доступності.

Висновки

1. Розроблена система аварійного відновлення вебресурсів у хмарному середовищі AWS здатна забезпечувати стабільну роботу ресурсу навіть у складних умовах, характерних для сучасних IT-інфраструктур, є досить гнучким рішенням та легко модифікується під різні бізнес вимоги. Вибір варіанта модифікації залежить від критичності даних та бюджетних обмежень. Якщо пріоритетом є мінімізація втрати даних (низький *RPO*) і час простою (низький *RTO*), то використання AWS DMS є кращим, хоча і дорожчим, рішенням. Проте, якщо економічна ефективність важливіша, а незначні втрати даних та час на відновлення є прийнятними, то підхід з відновленням з S3 є цілком достатнім.

2. Завдяки поєднанню локальних і хмарних ресурсів, автоматизації процесів перемикання та гнучкому механізму відновлення даних досягається значне зниження часу простою, мінімізація втрат

інформації, зменшення вартості утримання резервної інфраструктури, можливість адаптивного керування ресурсами, відсутність необхідності повного дублювання сервісів у хмарі та швидке масштабування у разі виникнення аварій.

3. Проведені експериментальні дослідження підтвердили, що запропонована система забезпечує відновлення працездатності у кілька разів швидше ніж класичні DR-підходи, а також є значно економнішою з погляду використання обчислювальних ресурсів, завдяки чому вона є ефективним рішенням для малих, середніх та великих підприємств, а також для державних установ або освітніх закладів, де критично важлива безперервність роботи систем.

4. Запропонована система аварійного відновлення вебресурсів добре масштабується під різні рівні критичності систем та може бути адаптована як для сервісів з низькими вимогами до доступності, так і для інфраструктур класу high-availability. Завдяки високій гнучкості системи придатна для організацій, які прагнуть уникнути значних витрат на повне дублювання інфраструктури, але потребують швидкого відновлення у разі аварій.

5. У подальших дослідженнях доцільним є розширення моделі DR шляхом інтеграції алгоритмів прогнозування збоїв на основі машинного навчання, оптимізації каналів реплікації даних, а також автоматичного вибору найефективнішого сценарію відновлення залежно від типу інциденту, навантаження системи та доступності хмарних ресурсів.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

[1] *TIC 2024 Hourly Cost of Downtime Report.* [Electronic resource]. Available: https://itic-corp.com/itic-2024-hourly-cost-of-downtime-report/?utm . Accessed: October 14, 2025.

[2] *The latest business continuity news from around the world. Survey shows that many organizations' disaster recovery plans lack maturity.* [Electronic resource]. Available: https://www.continuitycentral.com/news/business-continuity-news/7699-survey-shows-that-many-organizations-disaster-recovery-plans-lack-maturity . Accessed: October 15, 2025.

[3] P. Mell, and T. Grance, *The NIST Definition of Cloud Computing*, Special Publication (NIST SP), National Institute of Standards and Technology, Gaithersburg, MD, 2011. https://doi.org/10.6028/NIST.SP.800-145 . Accessed: November 12, 2025.

[4] *Amazon Web Services. AWS Disaster Recovery Guide.* [Electronic resource]. Available: https://docs.aws.amazon.com/pdfs/whitepapers/latest/disaster-recovery-workloads-on-aws/disaster-recovery-workloads-on-aws.pdf#disaster-recovery-options-in-the-cloud . Accessed: October 20, 2025.

[5] *Amazon Web Services. Route 53 health checks*. [Electronic resource]. Available: https://docs.aws.amazon.com/pdfs/Route53/latest/DeveloperGuide/route53-dg.pdf#dns-failover . Accessed on: October 20, 2025.

[6] Aureliu Zgureanu, "*The role of RPO and RTO in disaster recovery planning*. In: 30 years of economic reforms in the Republic of Moldova: economic progress via innovation and competitiveness," *The International Scientific Conference dedicated to the 30th Anniversary of the establishment of the Academy of Economic Studies of Moldova*, September 24th–25th, 2021, Chisinau. Chișinău: ASEM, 2022, vol. 3, pp. 221-232. ISBN 978-9975-155-66-3.

[7] *How to measure detection delay and repair time in equipment risk identification?* [Electronic resource]. Available: https://www.tencentcloud.com/techpedia/125791 . Accessed: October 22, 2025.

[8] *Amazon Web Services. Database Migration Service Documentation.* [Electronic resource]. Available: https://aws.amazon.com/dms . Accessed: November 23, 2025.

Рекомендована кафедрою обчислювальної техніки ВНТУ

Стаття надійшла до редакції 11.12.2025

Войцеховська Олена Валеріївна — канд. техн. наук, доцент, доцент кафедри обчислювальної техніки, e-mail: vojcexovska.o.v@vntu.edu.ua ;

Чехместрук Роман Юрійович — канд. техн. наук, доцент кафедри програмного забезпечення, e-mail: chekhroma@gmail.com ;

Марценюк Денис Вячеславович — студент факультету інформаційних технологій та комп'ютерної інженерії. Вінницький національний технічний університет, Вінниця;

Войцеховський Олександр Віталійович — студент факультету комп'ютерних наук та кібернетики. Київський національний університет імені Тараса Шевченка, Київ

O. V. Voitsekhovska[1]
R. Yu. Chechmestruk[1]
D. V. Martsenyuk[1]
O. V. Voitsekhovskyi[2]

Automatic Disaster Recovery System of Information Web Resources in a Cloud Environment

[1]Vinnytsya National Technical University;
[2]Taras Shevchenko National University of Kyiv

The work presents an analysis of modern approaches to disaster recovery (DR) of web applications in cloud environments and hybrid infrastructures. Main Disaster Recovery models are examined, such as Backup-and-Restore, Warm Standby, Multi-AZ, and Multi-Region, as well as their advantages and limitations in terms of implementation cost, recovery speed, and computational resource requirements. Special attention is paid to scenarios in which local servers are combined with Amazon Web Services cloud services, which make it possible to achieve a balance between cost efficiency and high fault tolerance.

Mechanisms for automating recovery processes are investigated, in particular the role of Amazon Route 53 Health Check, CloudWatch Alarm, SNS, and Lambda as components of the DR algorithm.

A system for automatic disaster recovery of web resources in a hybrid AWS environment is developed, which is based on automatic switching to the standby environment and selective recovery of only the critically important components of the system. Mathematical models for evaluating RTO and RPO indicators are constructed, taking into account time delays for failure detection, database recovery, application startup, and DNS route updates. Approaches to data recovery based on periodic backup to S3 and continuous replication using AWS Database Migration Service are developed. It is shown that traditional approaches requiring full duplication of infrastructure are resource-intensive and insufficiently flexible under conditions of dynamically changing availability of local resources.

A test web resource was deployed in the AWS cloud environment. A highly available infrastructure was designed using Amazon services, with all components created using the Infrastructure as Code approach with Terraform, which ensures repeatability, manageability, and the ability to quickly deploy solutions in different environments.

Based on the results of modeling and testing, a significant reduction in web resource downtime was established when applying the proposed disaster recovery system compared to traditional DR strategies.

Keywords: disaster recovery, AWS, web resource, hybrid infrastructure, replication, Route53, CloudWatch, Lambda, RTO, RPO.

Voitsekhovska Olena V. — Cand. Sc. (Eng.), Associate Professor, Associate Professor of the Chair of Computer Techniques, e-mail: vojcexovska.o.v@vntu.edu.ua ;

Chekmestruk Roman Yu. — Cand. Sc. (Eng.), Associate Professor of the Chair of Software Engineering, e-mail: chekhroma@gmail.com ;

Martsenyuk Denis V. — Student of the Department of Information Technologies and Computer Engineering;

Voitsekhovskyi Oleksandr V. — Student of the Department of Computer Science and Cybernetics